**Cassandra, MongoDB, Hypertable y HBase:**

Los más populares

* MongoDB (2.2)
  + Escrito en: C + +
  + Punto de principal: Retiene algunas propiedades amistosas de SQL. (Consulta, índice).
  + Licencia: AGPL (Drivers: Apache)
  + Protocolo: BSON
  + La replicación maestro / esclavo (failover automático con conjuntos de réplicas)
  + Sharding incorporado
  + Las consultas son expresiones javascript
  + Ejecutar arbitraria javascript funciones de servidor
  + Utiliza archivos asignados a la memoria para almacenamiento de datos
  + Rendimiento sobre las características
  + Journaling (con - journal) se gira best realizada el
  + En los sistemas de 32 bits, limitado a ~ 2,5 GB
  + Una base de datos vacía ocupa 192Mb
  + GridFS para almacenar grandes datos + metadatos (no en realidad un FS)
  + Tiene la indexación geoespacial
  + conocimiento del centro de datos
* Mejor usado: Si necesita consultas dinámicas. Si prefiere definir índices, no Map / Reduce funciones.
* Si usted necesita un buen rendimiento en una gran base de datos. Si querías CouchDB, pero sus datos también cambia mucho, llenando los discos.

Por ejemplo: Para la mayoría de las cosas que usted desea hacer con MySQL o PostgreSQL, pero habiendo columnas predefinidas realmente te retiene.

 habiendo Columnas predefinidas REALMENTE te retiene.

**Clones de Google de Bigtable**

* Cassandra (1.2)
  + Escrito en: Java
  + Dato: Best of BigTable y Dynamo
  + Licencia: Apache
  + Protocolo: Ahorro y binario personalizado CQL3
  + Ajustable concesiones para la distribución y reproducción (N, R, W)
  + Consulta de la columna, el rango de teclas (Requiere índices en cualquier cosa que desea buscar en)
  + Características BigTable similares: columnas, familias columna
  + Se puede utilizar como una tabla hash distribuida, con un lenguaje "tipo SQL", CQL (pero no ÚNETE!)
  + Los datos pueden tener vencimiento (ajustado en INSERT)
  + escribe puede ser mucho más rápido que lee (cuando lecturas son de disco-bound)
  + Map / Reduce posible con Apache Hadoop
  + Todos los nodos son similares, a diferencia de Hadoop / HBase
  + Replicación Cross-centro de datos
* Mejor usado: Cuando se escribe más de lo que lee (registro). Si cada componente del sistema debe estar en Java. ("Nadie es despedido por elegir cosas de Apache.")
* Por ejemplo: banca, la industria financiera (. Aunque no necesariamente para las transacciones financieras, pero estas industrias son mucho más que eso) Escribe son más rápidos que lee, por lo que un nicho natural es el análisis de datos.
* HBase (V0.92.0)
  + Escrito en: Java
  + Dato: Miles de millones de filas de X millones de columnas
  + Licencia: Apache
  + Protocolo: HTTP / REST (también Thrift)
  + Siguiendo el modelo de Google de BigTable
  + Utiliza HDFS de Hadoop como almacenamiento
  + map / reduce con Hadoop
  + predicado de consulta empuje hacia abajo a través de exploración lado del servidor y obtener filtros
  + Optimización de las consultas en tiempo real
  + Una pasarela Thrift alto rendimiento
  + HTTP soportes XML, protobuf y binario
  + JRuby basado (JIRB) shell
  + reinicio por turnos de los cambios de configuración y actualizaciones de menor importancia
  + El rendimiento de acceso aleatorio es como MySQL
  + Un grupo consiste en varios tipos diferentes de nodos
* Mejor usado: Hadoop es probablemente sigue siendo la mejor forma de ejecutar Map / Reduce en trabajos de grandes conjuntos de datos. Mejor si se utiliza la pila Hadoop / HDFS ya.
* Por ejemplo: los motores de búsqueda. El análisis de los datos de registro. Cualquier lugar en el que escanear grandes bidimensionales join-menos tablas, son un requisito.
* Hypertable (0.9.6.5)
  + Escrito en: C + +
  + Dato: A, HBase pequeño más rápido
  + Licencia: GPL 2.0
  + Protocolo: Thrift, biblioteca de C + +, o HQL shell
  + Implementa diseño BigTable de Google
  + Ejecutar en HDFS de Hadoop
  + Contiene su propio lenguaje "tipo SQL", HQL
  + Puede buscar por clave, por celular, o por los valores en las familias de columna.
  + La búsqueda puede limitarse a intervalos de teclas / columna.
  + Patrocinado por Baidu
  + Conserva los últimos N valores históricos
  + Las tablas están en espacios de nombres
  + map / reduce con Hadoop
* Mejor usado: Si necesita una mejor HBase.
* Por ejemplo: Lo mismo que HBase, ya que es básicamente un reemplazo: Los motores de búsqueda. El análisis de los datos de registro. Cualquier lugar en el que escanear grandes bidimensionales join-menos tablas, son un requisito.
* CouchDB (V1.2)

Escrito en: Erlang

Dato: la consistencia DB, facilidad de uso

Licencia: Apache

Protocolo: HTTP / REST

Replicación bidireccional (!)

continua o ad-hoc,

con detección de conflictos,

por lo tanto, la replicación maestro-maestro. (!)

MVCC - operaciones de escritura no bloquean lee

Las versiones anteriores de los documentos están disponibles

Crash-sólo el diseño (fiable)

Necesidades compactación de vez en cuando

Vistas: map / reduce incrustado

Vistas formato: listas y espectáculos

Validación de documentos del lado del servidor es posible autenticación posible

Actualizaciones en tiempo real a través de '\_changes' (!)

tratamiento de datos adjuntos

por lo tanto, CouchApps (js aplicaciones independientes)

Mejor usado: para acumular, en ocasiones cambiar los datos, en el que las consultas predefinidas se van a ejecutar. Lugares donde versionado es importante.

Por ejemplo, los sistemas CMS CRM. Replicación maestro-maestro es una característica especialmente interesante, que permite despliegues en múltiples sitios fáciles.